**Proceedings of the ASME 2018 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26–29, 2018, Quebec City, Quebec, Canada**

DETC2018-86213

A PROBLEM CLASS WITH COMBINED ARCHITECTURE, PLANT, AND CONTROL DESIGN APPLIED TO VEHICLE SUSPENSIONS

Daniel R. Herber, James T. Allison
University of Illinois at Urbana-Champaign
Industrial & Enterprise Systems Engineering
Urbana, IL 61801
Email: {herber1,jtalliso}@illinois.edu

ABSTRACT

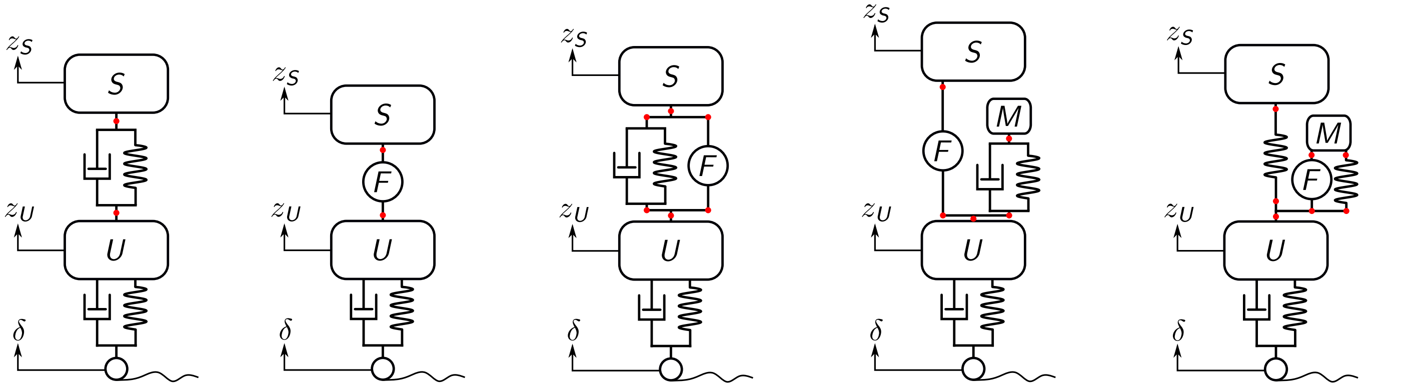
Here we describe a problem class with combined architecture, plant, and control design for dynamic engineering systems. The design problem class is characterized by architectures comprised of linear physical elements and nested co-design optimization problems employing linear-quadratic dynamic optimization. The select problem class leverages a number of existing theory and tools and is particularly attractive due to the symbiosis between labeled graph representations of architectures, dynamic models constructed from linear physical elements, linear-quadratic dynamic optimization, and the nested co-design solution strategy. A vehicle suspension case study is investigated and a specifically constructed architecture, plant, and control design problem is described. The result was the automated generation and co-design problem evaluation of 4,374 unique suspension architectures. The results demonstrate that changes to the vehicle suspension architecture can result in improved performance, but at the cost of increased mechanical complexity. Furthermore, the case study highlights a number of challenges associated with finding solutions to the considered class of design problems.

1 INTRODUCTION

The design of vehicle suspensions has been of considerable interest ever since the invention of the automobile. A suspension transfers forces in the system to provide a smooth ride for the passenger and good handling characteristics among other objectives. Fundamentally, it is a type of vibration isolator [1–3]. Different types of suspensions can typically be classified into the three categories of passive [1–6], semi-active [3, 7–9], or active [2, 3, 10–13] depending on the external energy flow into the system. There has been considerable research interest in analyzing and optimizing all types of suspensions, particularly utilizing tools from dynamics and controls. However, much of this research has focused on a select few canonical suspensions such as the ones shown in Figs. 1a–1d. Here we will consider architecture changes in the suspensions, i.e., different components connected in new ways such as the suspension shown in Fig. 1e.

In one-dimensional suspension systems, an unsprung mass U with position z_U is connected to the road profile δ, typically through a parallel spring and damper that captures the tire dynamics. There is also a sprung mass S with position z_S. There will exist some mechanical path (known as the suspension) between S and U. The performance of a suspension system is then evaluated by observing the dynamic behavior of both the sprung and unsprung masses with respect to road profile variations, and several candidate performance metrics based on this behavior exist. This is a commonly-used simplified treatment of the kinematics, but can only capture the performance metrics to certain degree of accuracy.

In addition to potential architecture design changes, we will consider design elements in both the physical (or plant) and control design domains. Here plant variables are time-independent quantities that relate to the physical embodiment of the suspension such as the stiffness of the springs or the damping coefficient for the dampers. The control design is associated with the actuator force trajectories (e.g., the force produced by component F in Fig. 1b) which are time-varying inputs to the system that can regulate the dynamic behavior of the system.



(a) Canonical passive [4,6]. (b) Pure active [2,3]. (c) Canonical active [5,10–13]. (d) Active with dynamic absorber [2,3]. (e) An alternative architecture.

FIGURE 1: Some vehicle suspension architectures.

This level of design freedom leads to a complex, challenging design problem that is typically not treated in a systematic fashion in engineering design practice. Due to the large scope of the proposed design study, many simplifying assumptions must be made to keep the problem tractable, but we need a certain level of veracity such that it is still possible to reveal the desired design insights. This motivates the use of a particular problem class of combined architecture, plant, and control problems that has sufficiently developed theory and tools that can be leveraged to provide relatively efficient solutions. This design problem class selected here is characterized by architectures comprised of *linear physical elements* (LPEs) and nested co-design optimization problems employing *linear-quadratic dynamic optimization* (LQDO). The problem class will be fully defined in the following section.

It has been observed by suspension design experts that even simple suspension models and studies have had a "profound impact on the practical implementations some 15–20 years later [3]". In addition, suspension design problems with similar design fidelity have proven to be an interesting area for research in combined plant and control design or co-design [10–12]. These co-design studies considered a single architecture, which is typically the canonical active system in Fig. 1c. It is important to note that the purpose of the early-stage studies proposed in this article is not to supplant the detailed, rigorous previous research, but rather seeks to identify new architectures that could be investigated in the same level of detail that the few canonical architectures have received.

The remainder of this article is organized as follows. Section 2 discusses some important considerations when dealing with combined architecture, plant, and control design problems. Section 3 outlines the considered problem class and proposed solution approach. Section 4 describes the set of candidate vehicle suspension architectures. Section 5 details the combined architecture, plant, and control vehicle suspension design problem formulation. Section 6 presents the results of the case study and Sec. 7 provides the conclusions.

2 CONSIDERATIONS FOR A PROBLEM CLASS WITH COMBINED ARCHITECTURE, PLANT, AND CONTROL DESIGN

Since the proposed level of design freedom leads to a complex, challenging design problem, we first list some important considerations. Addressing these comments will eventually facilitate the study of different vehicle suspensions.

1. Meaningful candidate architectures can readily be generated; this is typically handled by defining suitable components that can be connected together in meaningful ways.
2. Given some representation of an architecture, a model can be automatically constructed that captures its dynamic behavior.
3. The components are defined at an acceptable level with respect to modeling and design while preferring computationally inexpensive embodiments.
4. The evaluation of a candidate architecture's performance is possible and efficient.
5. Since we are considering dynamic systems, it is preferred that the generated model provides an explicit derivative function.

Now, we can discuss how systems with LPEs, LQDO, and nested co-design at least partially address these considerations.

2.1 Linear Physical Elements

A useful framework for describing LPEs is bond graph modeling with power port nodes (or simply power nodes) [14]. Power nodes are characterized by constitutive parameters and follow some constitutive relation (typically a fundamental physical law). They can be classified as source nodes (Se, Sf), storage nodes (C, I), resistive nodes (R), reversible transducers (TF, GY), and

TABLE 1: Some bond graph modeling analogies.

	Linear Mechanical			
Label	Intuitive	Topology Preserving	Electrical	Hydraulic
Se	Force	Velocity	Voltage	Pressure
Sf	Velocity	Force	Current	Volume
C	$1/K$	M	C	C_f
I	M	$1/K$	I	I_f
R	B	$1/B$	R	R_f

junction nodes (*0*, *1*). Some analogies for these power nodes in different energy domains are listed in Table 1. An example of a *TF* (transformer) is a lever, gear, or hydraulic cylinder. The *GY* (gyrator) typically describes the conversion between energy domains, such as with an electrical DC motor or mass accelerometer. The *0*-junction is analogous to Kirchhoff's current law and the *1*-junction is analogous to the voltage law in electrical systems. With these elements, many (multi-domain) engineering systems with mechanical, electrical, hydraulic, magnetic, etc. elements can be modeled to some degree. For more details on bond graph modeling, see Refs. [14–16].

The key property for systems represented by bond graphs with linear time-invariant elements is that the equations of motion can be represented as a *linear descriptor model* [17, 18]. If we denote the set of all constitutive parameters for a particular bond graph as $\boldsymbol{p}$, the dynamic model is of the form:

$$\boldsymbol{E}(\boldsymbol{p})\dot{\boldsymbol{\xi}} = \boldsymbol{A}(\boldsymbol{p})\boldsymbol{\xi} + \boldsymbol{B}(\boldsymbol{p})\boldsymbol{s} \tag{1}$$

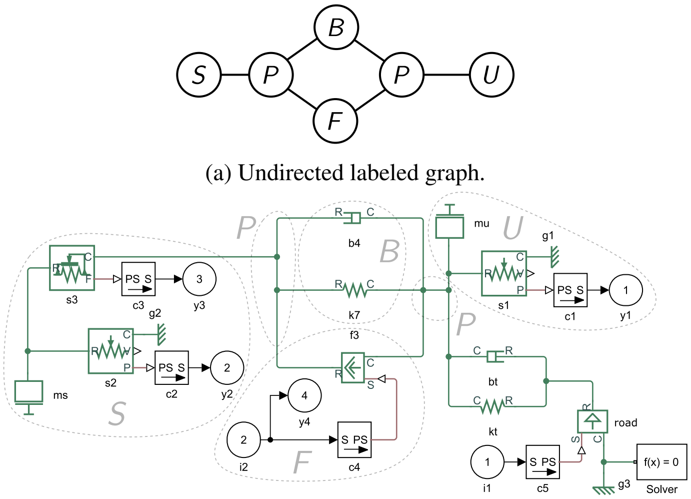
where $\boldsymbol{\xi}$ are the states and $\boldsymbol{s}$ are the sources. The matrix $\boldsymbol{E}$ is invertible if there are no algebraic loops in the system [14, 17, 18]. Here we assume that all algebraic loops are appropriately removed (e.g., see Ref. [17] or the modeling approach used below) so that we have an explicit first-order ordinary differential equation (ODE).

We will also utilize the notation of outputs $\boldsymbol{y}$ to capture certain dynamic quantities in the architecture. As long as the desired outputs (e.g., acceleration of *S* or position of the *U*) are a linear combination of the states and sources, then we can obtain a linear output equation in the standard state-space representation:

$$\boldsymbol{y} = \boldsymbol{C}(\boldsymbol{p})\boldsymbol{\xi} + \boldsymbol{D}(\boldsymbol{p})\boldsymbol{s} \tag{2}$$

Now, the architecture design decisions will include what power nodes to include in the system and their connections, satisfying consideration 1. Some of the constitutive parameters will be the plant design variables, such as the stiffness coefficient of the spring component. The control decisions will come in the form of certain source types. The sources may also be used to add various disturbances to the system. Since bond graph models and optimization with respect to constitutive parameters and source inputs are commonly accepted, we can satisfy consideration 3 to a certain degree.

Utilizing architectures consisting of LPEs permits the use of a number of available tools for automated model creation, which are needed to satisfy consideration 2. For general bond graph models, tools such as MODELICA [19] and MTT [20] are available. How, it was found to be challenging to obtain the explicit ODE representing the dynamics (consideration 5). To obtain the desired models, code was developed that takes the chosen graph representation of the architecture and constructs the appropriate SIMSCAPE™/SIMULINK® model with all components placed and input/output nodes defined [21]. This is illustrated in Fig. 2 for a canonical active architecture. With generated SIMSCAPE™/SIMULINK® model, the derivative and output functions can be generated through linearization[1], which exactly represents the considered architectures [22].



(a) Undirected labeled graph.

(b) SIMSCAPE™/SIMULINK® model.

FIGURE 2: Different representations of the canonical active suspension architecture in Fig. 1c.

2.2 Linear-Quadratic Dynamic Optimization

Consider the following finite horizon, continuous time dynamic optimization problem with linear constraints and a quadratic objective form:

$$\min_{\boldsymbol{u}} \quad \int_{t_0}^{t_f} \left(\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\xi} \end{bmatrix}^{\mathsf{T}} \boldsymbol{L} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\xi} \end{bmatrix} + \boldsymbol{l}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\xi} \end{bmatrix} + c_L \right) dt + \cdots \tag{3a}$$

$$\begin{bmatrix} \boldsymbol{\xi}(t_0) \\ \boldsymbol{\xi}(t_f) \end{bmatrix}^{\mathsf{T}} \boldsymbol{M} \begin{bmatrix} \boldsymbol{\xi}(t_0) \\ \boldsymbol{\xi}(t_f) \end{bmatrix} + \boldsymbol{m}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\xi}(t_0) \\ \boldsymbol{\xi}(t_f) \end{bmatrix} + c_M \tag{3b}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}} = \boldsymbol{A}\boldsymbol{\xi} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d} \tag{3c}$$

$$\boldsymbol{C}_1\boldsymbol{\xi} + \boldsymbol{C}_2\boldsymbol{u} \le \boldsymbol{C}_3 \tag{3d}$$

$$\boldsymbol{\phi}_1\boldsymbol{\xi}(t_0) + \boldsymbol{\phi}_2\boldsymbol{\xi}(t_f) \le \boldsymbol{\phi}_3 \tag{3e}$$

where $t \in [t_0, t_f]$ is the time continuum, $\boldsymbol{u}$ are the open-loop con-

[1] However, the linearization procedure is relatively expensive at ~0.5 s, and must be run for every change in the plant variables. More efficient options are being investigated.

trols (OLCs), and the matrices in Eqns. (3a), (3c), and (3d) can be time varying. This particular optimization problem, known as an LQDO problem, can be approximated as a finite-dimensional quadratic program (QP) using direct transcription methods [23]. Under certain conditions, the QP is convex and efficient to solve for even large systems and the global optimal solution for $\boldsymbol{u}$ (and $\boldsymbol{\xi}$) with respect to the discretization is guaranteed. Therefore, utilizing the LQDO form helps satisfy consideration 4. While Prob. (3) is only a subclass of general LQDO, it is suitable to represent the problems of interest here and the more general form presented in Ref. [23] is what we define as the LQDO problem class.

2.3 Nested Co-Design Solution Strategy

Consider the following general co-design problem:

$$\min_{\boldsymbol{x}_p, \boldsymbol{x}_c} \quad \Psi(\boldsymbol{x}_p, \boldsymbol{x}_c) \tag{4a}$$

$$\text{subject to:} \quad \boldsymbol{g}_o(\boldsymbol{x}_p) \leq \boldsymbol{0}, \; \boldsymbol{g}_i(t, \boldsymbol{x}_p, \boldsymbol{x}_c) \leq \boldsymbol{0} \tag{4b}$$

where $\boldsymbol{x}_p$ are the plant variables, $\boldsymbol{x}_c$ are the control variables, $\Psi(\cdot)$ is the general objective function, and $\boldsymbol{g}$ represent general constraints. Constraints $\boldsymbol{g}_o$ capture the time independent, plant-only constraints while $\boldsymbol{g}_i$ captures all time, control, and state dependent constraints including the dynamics.

Two common methods for solving co-design problems are the simultaneous and nested solution strategies [10, 24, 25]. In the simultaneous approach, we seek a feasible minimizer while simultaneously modifying both $\boldsymbol{x}_p$ and $\boldsymbol{x}_c$. To define the nested approach, we first consider the set of feasible control designs given candidate plant $\boldsymbol{x}_p^\dagger$ with respect to $\boldsymbol{g}_i$:

$$\Omega_i(\boldsymbol{x}_p^\dagger) = \left\{ \boldsymbol{x}_c \; : \; \boldsymbol{g}_i(t, \boldsymbol{x}_p^\dagger, \boldsymbol{x}_c) \leq \boldsymbol{0} \right\} \tag{5}$$

Then the nested strategy proceeds through a two-level optimization scheme where an *outer loop* optimizes with respect to the plant variables and an *inner loop* optimizes with respect to the control variables for a given plant design:

$$\boldsymbol{x}_p^* = \arg\min_{\boldsymbol{x}_p} \left\{ \Psi(\boldsymbol{x}_p, \mathcal{I}(\boldsymbol{x}_p)) \; : \; \boldsymbol{g}_o(\boldsymbol{x}_p) \leq \boldsymbol{0} \right\} \quad \text{(outer)} \tag{6a}$$

$$\mathcal{I}(\boldsymbol{x}_p^\dagger) = \arg\min_{\boldsymbol{x}_c} \left\{ \Psi(\boldsymbol{x}_p^\dagger, \boldsymbol{x}_c) \; : \; \boldsymbol{x}_c \in \Omega_i(\boldsymbol{x}_p^\dagger) \right\} \quad \text{(inner)} \tag{6b}$$

where the mapping $\mathcal{I} : \boldsymbol{x}_p \mapsto \boldsymbol{x}_c$ must exist for all considered values of $\boldsymbol{x}_p$ for the simultaneous and nested approaches to be equivalent [24, 25].

The key advantage of the nested approach is that different optimization methods can be utilized in the inner/outer loops. Therefore, the partitioning of the inner/outer loops can be justified if there are efficient methods that cannot be leveraged using the simultaneous form. In fact, a co-design problem with LPEs and OLCs variables can be formed such that the inner-loop problem is an LQDO problem (and the simultaneous form would need to be at least bilinear). This is due to the fact that for fixed values of $\boldsymbol{p}$, Eqn. (1) with invertible $\boldsymbol{E}$ is in the appropriate form for Eqn. (3c) in LQDO. A number of studies have shown the benefits of the nested strategy when an efficient inner-loop solution method is available [25–27], and utilizing nested co-design helps satisfy consideration 4. Additionally, since co-design problems are potentially nonconvex, we can now focus our global search efforts only in the outer loop with respect to the plant variables since the inner loop is convex.

3 PROBLEM CLASS DEFINITION AND SOLUTION APPROACH

We now will formally define the problem class of interest and the proposed solution approach.

3.1 Problem Class Definition

We would like to solve architecture design problems posed as combinatorial optimization problems of the following form:

$$\min_{\boldsymbol{x}_a} \quad \Psi = \Psi_a(\boldsymbol{x}_a) + \Psi_d\left(\mathcal{X}_p(\boldsymbol{x}_a), \mathcal{X}_c(\boldsymbol{x}_a)\right) \tag{7a}$$

$$\text{subject to:} \quad \mathcal{X}_a(\boldsymbol{x}_a) = a \in \mathcal{F}_a \tag{7b}$$

$$\boldsymbol{x}_a \in \mathcal{F}_x \tag{7c}$$

where $\boldsymbol{x}_a$ represents discrete architecture design variables, $\mathcal{X}_a : \boldsymbol{x}_a \mapsto a$ is a mapping between the $\boldsymbol{x}_a$ and the architecture a, $\mathcal{F}_x$ is the set of feasible architecture design variables, and $\mathcal{F}_a$ is the set of feasible architectures. Ψ_a is the architecture-only objective function while Ψ_d is the general co-design objective function that includes dependence on the plant and control design. This dependence is represented by the mapping functions $\mathcal{X}_p : \boldsymbol{x}_a \mapsto \boldsymbol{x}_p$ and $\mathcal{X}_c : \boldsymbol{x}_a \mapsto \boldsymbol{u}$ between the architecture and the plant $\boldsymbol{x}_p$ and OLC design variables. Typically, we ensure that $\boldsymbol{x}_a$ is finite and there is countable set of feasible combinations. This will lead to $\mathcal{F}_a$ being a discrete, countable set, typically with fewer elements than $\mathcal{F}_x$ as a consequence of isomorphisms, i.e., multiple values of $\boldsymbol{x}_a$ producing the same a [28].

A fair comparison between architecture candidates requires knowledge of the best possible performance for each candidate architecture. To determine the value of Ψ_d we must solve a suitable co-design problem of the following form:

$$\min_{\boldsymbol{x}_p^{(i)}, \boldsymbol{u}^{(i)}} \quad \Psi_d = \int_{t_0}^{t_f} \mathcal{L}^{\mathcal{N}}\left(t, \boldsymbol{y}, \boldsymbol{x}_p^{(i)}\right) dt + \cdots \tag{8a}$$

$$\mathcal{M}^{\mathcal{N}}\left(\boldsymbol{y}(t_0), \boldsymbol{y}(t_f), \boldsymbol{x}_p^{(i)}\right) \tag{8b}$$

$$\text{subject to:} \quad \left[\dot{\boldsymbol{\xi}} = \boldsymbol{f}^{\mathcal{N}}\left(t, \boldsymbol{\xi}, \boldsymbol{u}, \boldsymbol{x}_p\right)\right]^{(i)} \tag{8c}$$

$$\boldsymbol{g}^{\mathcal{N}}\left(t, \boldsymbol{y}, \boldsymbol{x}_p^{(i)}\right) \leq \boldsymbol{0} \tag{8d}$$

$$\text{where:} \quad \boldsymbol{y} = \boldsymbol{y}\left(t, \boldsymbol{\xi}^{(i)}, \boldsymbol{u}^{(i)}, \boldsymbol{x}_p^{(i)}\right) \tag{8e}$$

where $\square^{(i)}$ indicates a problem formulation element appropriate for the ith candidate architecture from Prob. (7), $\boldsymbol{y}$ are the architecture-dependent outputs in Eqn. (2), and the superscript

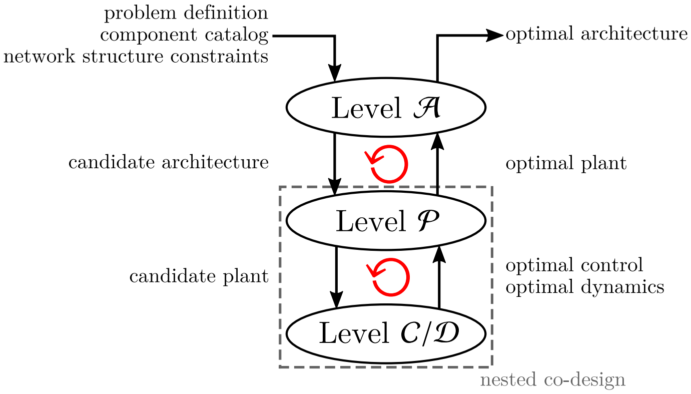


FIGURE 3: The proposed trilevel solution strategy for combined architecture, plant, and control design of a dynamic system.

$\mathcal{N}$ indicates a particular problem class that the problem elements must be in. Here we consider $\mathcal{N}$ as the LQDO-amenable co-design problem class based on the considerations in Sec. 2.

Definition 1. *A co-design problem is said to be an LQDO-amenable co-design problem if for fixed values of $\boldsymbol{x}_p$, all problem elements are within the LQDO problem class in Sec. 2.2.*

To ensure that Eqn. (8c) satisfies these conditions, the models of the architectures must be constructed from LPEs (see Sec. 2.1). With the problem class defined, we can now describe the trilevel solution strategy.

3.2 A Trilevel Solution Approach

Consider the following definition of an optimal solution to a combinatorial optimization problem.

Definition 2. *Solution $\boldsymbol{x}_a^* \in \mathcal{F}_x$ and $a^* \in \mathcal{F}_a$ is an optimum solution of Prob. (7) if $\Psi(\boldsymbol{x}_a^*) \leq \Psi(\boldsymbol{x}_a)$ for all $\boldsymbol{x}_a \in \mathcal{F}_x$ and $X_a(\boldsymbol{x}_a) = a \in \mathcal{F}_a$.*

Unlike some combinatorial optimization problems, such as the traveling salesman, knapsack, assignment, or vertex coloring problems [29], there are limited exact or approximation algorithms that are suitable for the selected problem class. This might be due to the fact that although architectures are defined by graph-theoretic concepts, their performance is not. Until such solution approaches exist, the only way to guarantee that you find the optimal architecture is to evaluate all feasible architectures. However, such an approach is only suitable for problems of a certain size, and this idea is discussed further in Sec. 3.3.

The proposed solution approach treats each design domain modularly to leverage existing theory and tools that provide relatively efficient solutions and address the aforementioned considerations. This leads to the trilevel solution approach, where each design domain is a level, and is illustrated in Fig. 3. Each one of the three levels is now described.

3.2.1 Architecture Design: Level $\mathcal{A}$.

The topmost level is responsible for taking the problem definition, a user-defined component catalog, and network structure constraints and providing candidate architectures for the other levels.

Here we will use undirected labeled graphs to represent candidate architectures [28, 29]. The different labels in the graphs will correspond to the power nodes in Sec. 2.1 or superelements (some subgraph of power nodes) [14–16, 23]. There is a strong resemblance between the equivalent bond graph or block-diagram model (such as a SIMSCAPE™/SIMULINK® model). For example, compare Figs. 2a and 2b for a specific suspension architecture. Some advantages of using a labeled graph representation are that labels can be used to represent a variety of concepts and existing graph theory and tools can be utilized.

Working with undirected labeled graphs, the brute-force generation procedure described in Ref. [28] will be used to produce an enumeration (or complete listing) of all the feasible architectures in the considered feasible space. This approach utilizes a perfect matching-inspired algorithm where all ports (or potential connections) of every component (or vertex in the graph) are connected to exactly one other port [28, 30]. In the worst case scenario, the growth of the number of graphs (architectures) is $(N-1)!!$ where N is the total number of ports and $(\cdot)!!$ represents the double factorial function. However, this bound is extremely conservative as many of the generated graphs are isomorphic (not unique) or do not satisfy network structure constraints (NSCs). Satisfaction of all NSCs defines the feasibility for a particular graph and the set $\mathcal{F}_a$ [28, 31]. Many enhancements to the original approach in Refs. [28, 32] have been made in Ref. [33], further leveraging the structure of the enumeration task and allowing reasonably large graph structure spaces to be enumerated.

The required information for this approach is the following designer-defined elements:

- C is the label sequence representing distinct component types
- P is a vector indicating the number of ports for each component type
- $(R_{\min}, R_{\max})$ are a vectors indicating the minimum/maximum number of replicates for each component type; R is the collection of both vectors

The collection (C, R, P) is termed the component catalog and defines $\mathcal{F}_x$, where $\boldsymbol{x}_a$ are different perfect matchings.

3.2.2 Plant Design: Level $\mathcal{P}$.

The next level takes the candidate architecture and performs the outer-loop co-design tasks for the plant design [25]. The automated model generation procedure described in Sec. 2.1 and shown in Fig. 2 creates a suitable model. Then the appropriate optimization problem in the form of Prob. (6a) is automatically created and solved. Since this type of problem can be nonconvex, global search algorithms are utilized to help improve the confidence of finding the true optimal solution.

3.2.3 Control Design and Dynamics: Level $\mathcal{C}/\mathcal{D}$.

This deepest level takes the model and candidate plant and formulates the appropriate LQDO problem. Note that the number of times Level $\mathcal{C}/\mathcal{D}$ is solved is much greater than Level $\mathcal{P}$, which is also solved for many times in Level $\mathcal{A}$ for different candidate architectures. Successfully solving a Level $\mathcal{C}/\mathcal{D}$ problem provides both the optimal control and dynamics, thus, providing a evaluation of Ψ_d. Here we will utilize the Matlab code available at Ref. [34] for solving the LQDO problem using direct transcription methods [23]. The code utilizes a structure-based description of the optimization problem, which makes it relatively straightforward to handle a varying number of states and controls along with the output definitions [23].

3.3 Comments

The nonlinear constitutive relations can be captured by the enumeration approach in Ref. [32] assuming models can be appropriately constructed from undirected edges. A key consideration is still the need for an explicit derivative function (consideration 5). Constructing a simulation-based model is typically not satisfactory for many of the numerical methods needed to efficiently solve the co-design problem [35]. An example of an alternative graph-based modeling approach is presented in Ref. [36] and is applicable to fluid-based thermal systems. Given a graph-based architecture specification, the explicit bilinear derivative function can be constructed (so these models do not fit within the purview of LQDO). Optimica is another tool that may facilitate modeling and optimization for the problem class of interest [37].

If we instead require general co-design problems for Prob. (8) rather than LQDO-amenable co-design problems, the simultaneous co-design strategy could be more efficient in some cases, and may avoid potential infeasibility issues associated with $\mathcal{I}$ [25]. There are a number of tools available to solve the resulting nonlinear dynamic optimization such as Gpops-ii [38]. There are also many interesting architecture design problems without plant or control systems, such Levels $\mathcal{A}/\mathcal{P}$ for passive analog circuit synthesis [23] or Levels $\mathcal{A}/(\mathcal{C}/\mathcal{D})$ for fluid-based thermal circuits [39].

There are also alternatives to enumeration in Level $\mathcal{A}$. Semi-optimal, heuristic-based methods for solving the combinatorial problem have included graph grammars [40], A-design [41], evolutionary computation [42,43], simulated annealing [44], or machine learning techniques [45]. With some of these approaches, both architecture and plant design can be solved for simultaneously [42], although some argue in favor of the nested approach for certain problems [43]. However, to the authors' knowledge, none of these have attempted simultaneous architecture and control design or dynamic optimization with direct transcription. Going beyond enumeration is essential for attempting to find solutions to large combined architecture, plant, and control design problems.

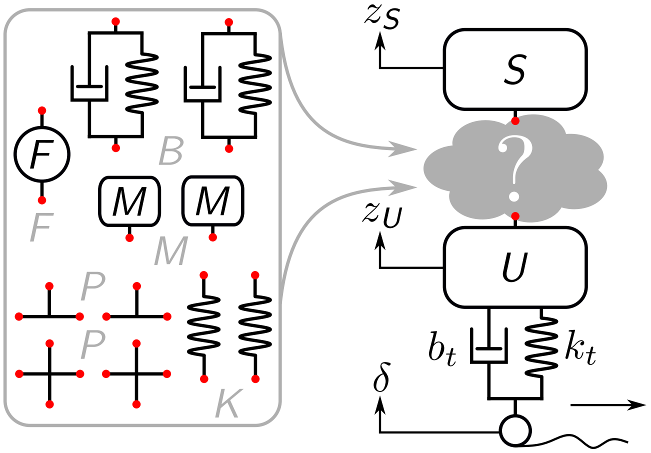


FIGURE 4: Vehicle suspension architecture design domain.

4 SPECIFICATION OF THE CANDIDATE VEHICLE SUSPENSION ARCHITECTURES

Here we will use undirected labeled graphs to represent candidate vehicle suspension architectures as described in Sec. 3.2.1 [28].

4.1 Component Catalog Specification

The component catalog will be the similar to the one used in Refs. [28,33]:

$$C = \{S\ U\ M\ K\ B\ F\ P\ P\} \tag{9a}$$

$$R_{\min} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{9b}$$

$$R_{\max} = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 2 & 2 \end{bmatrix} \tag{9c}$$

$$P = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 4 \end{bmatrix} \tag{9d}$$

The location for where these components may be included in the suspension system is represented in Fig. 4. Each of the component types fits within the bond graph modeling paradigm: $\{S, U, M\}$ are I-type storage nodes, K is a C-type storage node, B is a subsystem containing a spring and damper (R-type node) in parallel (see Fig. 4), and F is an Se-type effort source. The remaining component types represent 1-junctions with differing numbers of ports.

The selection of the component types and their model was based on observations on the components that have been commonly used in different suspensions. The required components are (S, U, F), so we are considering active suspensions only. The maximum number of replicates for the other component types was selected based on what is reasonable for enumeration as well as being able to explore a reasonable space of novel suspension architectures.

4.2 Network Structure Constraints

All the NSCs for this problem are listed in Ref. [28]. Here we summarize only a few of the more important ones:

- Each included component must have unique edges in the graph (i.e., no multi-edges).
- A mechanical path between the U and S must be present and this path cannot only contain P-type components.
- Certain component types should not be connected together (e.g., $U-S$, $K-K$, or $P-P$).
- Only consider a specific ordering of the series connections because permuting them results in an equivalent model (e.g., $K-B$ and $B-K$ in series are physically equivalent).
- Component F must be connected to two different P except for the canonical active suspension to prevent some modeling issues (see Sec. 5.3), and this NSC could be removed when the internal positioning constraints are properly included.

4.3 Architecture Cost Metric

The architecture-only objective function term considered here is the sum of the additional physical components (i.e., all components except S, U, and P):

$$\Psi_a = w_a N_c = w_a \left(n_M + n_K + 2n_B + n_F\right) \tag{10}$$

where w_a is the weighting coefficient. This is just one metric for attempting to account for the complications that can occur from an increased mechanical complexity of the system (e.g., monetary cost, reliability, and other design challenges).

5 ARCHITECTURE-DEPENDENT CO-DESIGN PROBLEM

The co-design problem can be posed through the outputs, dynamics, objective, and constraints. The forms of these problem elements are consistent across the different candidate architectures, but their specific realization will be architecture dependent.

5.1 Dynamic and Output Equations

A number of outputs are needed to realize the desired co-design problem:

$$\boldsymbol{y} = \left[z_U,\, z_S,\, \ddot{z}_S,\, F,\, z_F,\, \boldsymbol{z}_B^{\mathsf{T}},\, \boldsymbol{z}_K^{\mathsf{T}}\right]^{\mathsf{T}} \tag{11}$$

namely, the unsprung mass position, sprung mass position, sprung mass acceleration, actuator force, actuator position, $n_b^{(i)}$ B-type component positions, and $n_k^{(i)}$ K-type component positions. The two sources or inputs are the force trajectory of the actuator F and the road velocity profile $\dot{\delta}$. Then the dynamics and output equations generated from the linearization of the programmatically-created model as described in Sec. 2.1 will have the following form:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}\boldsymbol{\xi} + \boldsymbol{B}_f F + \boldsymbol{B}_\delta \dot{\delta} \tag{12a}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{\xi} + \boldsymbol{D}_f F + \boldsymbol{D}_\delta \dot{\delta} \tag{12b}$$

noting that the matrices are dependent on the plant variables and problem parameters.

5.2 Objective

The co-design objective function will be the sum of several common performance metrics:

$$\Psi_d = \int_{t_0}^{t_f} \left(w_1 \left(y_1 - \delta\right)^2 + w_2 y_3^2 + w_3 y_4^2\right) dt \tag{13}$$

where the term $w_1 \left(y_1 - \delta\right)^2$ captures the *handling* performance, $w_2 y_3^2$ represents the passenger *comfort*, and $w_3 y_4^2$ control *effort* objective (see Refs. [10–12]). Combining Eqns. (12b) and (13) results in an LQDO objective function [23].

5.3 Constraints

Next, the states of the system are initialized to their zero equilibrium position with the following simple bound constraint:

$$\boldsymbol{\xi}^{(i)}(t_0) = \boldsymbol{0} \tag{14}$$

To ensure that the separation between the sprung and unsprung masses remains tolerable, the following rattlespace constraint is necessary [3, 4, 10, 12, 13]:

$$|r| := |y_2 - y_1| \le r_{\max} \tag{15}$$

This constraint can be converted into two linear constraints [23]. The LQDO problem class can readily handle linear inequality constraints unlike other solution strategies [12, 23]. Similarly, stroke constraints are included on the 2-port components $\{F, B, K\}$ to ensure that their travel is within a tolerable level:

$$|y_5| \le s_{\max} \tag{16a}$$

$$|y_{5+j}| \le s_{\max} \qquad j = 1, \ldots, n_b^{(i)} + n_k^{(i)} \tag{16b}$$

Internal positioning constraints (i.e., constraints that ensure that all internal positions between U and S are properly ordered and between these masses) should also be included, but is currently left as future work. These constraints are essential when F is included in an architecture since the model for F ignores the velocity across it. To facilitate an easier visualization of the stroke constraints, we denote the largest magnitude all the stroke constraints as $S_{\max}(t)$.

All the previous constraints are necessary for the inner-loop problem using the nested co-design strategy. The outer-loop specific plant constraints are simple bounds on the linear coefficients [5, 10, 11]:

$$m_{\min} \le m_j \le m_{\max} \qquad j = 1, \ldots, n_m^{(i)} \tag{17a}$$

$$b_{\min} \le b_j \le b_{\max} \qquad j = 1, \ldots, n_b^{(i)} \tag{17b}$$

$$k_{\min} \le k_j \le k_{\max} \qquad j = 1, \ldots, n_k^{(i)} \tag{17c}$$

TABLE 2: Co-design problem parameters.

Parameter	Value	Parameter	Value
t_0	0 s	t_f	3 s
$m_{\min}$	10^{-2} kg	$m_{\max}$	10^{1} kg
$b_{\min}$	10^{2} Ns/m	$b_{\max}$	10^{5} Ns/m
$k_{\min}$	10^{2} N/m	$k_{\max}$	10^{6} N/m
$r_{\max}$	0.04 m	$s_{\max}$	0.04 m
m_U	65 kg	m_S	325 kg
w_1	10^{5}	k_t	232.5×10^{3} N/m
w_2	0.5	b_t	0 Ns/m
w_3	10^{-5}		

where the subscripts (m,b,k) indicate the additional mass, damper, and spring plant variables in the candidate architecture. While we are only using simple bounds in for the plant-specific constraints, a more comprehensive plant design formulation can be included in an LQDO-amenable co-design problem [23, 26]. For example, we could instead design the springs based on their geometry [12, 46].

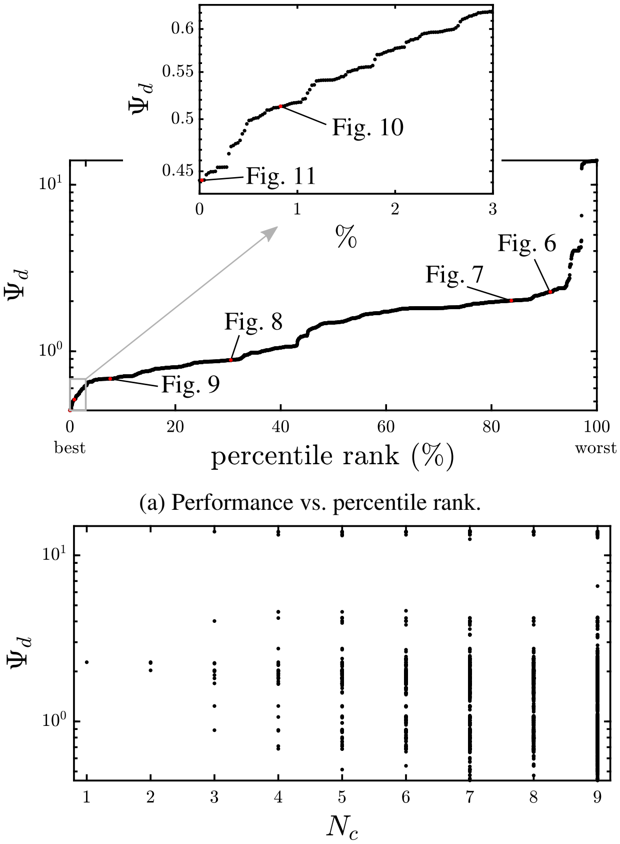
6 CASE STUDY RESULTS

The problem parameters used in this study are shown in Table 2 and many are based the study in Ref. [12]. A rough road input from Refs. [6, 12] was used. With the component catalog and NSCs described in Sec. 4, there are 4,374 active suspension candidate architectures. For each of these suspension candidates, the optimal performance was determined by solving an automatically generated co-design problem specific to the candidate architecture. The Matlab code is available at Ref. [21].

The results are summarized in Fig. 5 for 4,351 architectures. In Fig. 5a, we see a large distribution for the performance results, including a select few that perform extremely poorly. For sixteen of the candidates, a feasible co-design solution could not be determined, but a more exhaustive search may lead to feasible solutions for these architectures. For seven of the architectures, the internal position constraints mentioned in Sec. 5.3 (which were not included in these results) were violated to an unacceptable degree producing unrealistically low values for Ψ_d, and therefore were removed from the results. Now, shown in Fig. 5b are the architectures stratified by the mechanical complexity metric in Eqn. (10). In general, we observe that increasing the number of components can improve performance, but this trend only holds up to a certain point where additional complexity does not lead to better performance. However, there still are a large number of candidate architectures with large N_c that perform poorly. The complete set of results can be used to help determine if the increased mechanical complexity of a new candidate architecture is worth the performance improvement.

The results for a few candidate suspensions are shown in Table 3 and Figs. 6–11, including the pure active and canonical active suspensions. The best performing architecture with $N_c = 3$ is shown in Fig. 8, and is decidedly different than the canonical active architecture having the same number of components. In Fig. 9, the best performing architecture with $N_c = 4$ is shown. It is comprised out of two springs and one additional mass and resembles the dynamic absorber architecture in Fig. 1d. The primary difference between these architectures is that the force actuator is attached to the additional mass rather than between U and S. Finally, the best architectures for $N_c = 5$ and $N_c = 7$ are shown in Figs. 10 and 11, respectively. The best architecture for $N_c = 6$ is not shown as there is no performance improvement over a less mechanically complex architecture.

Comparing the performance measures, the value of Ψ_d is much lower for the other architectures than for the canonical active suspension (4.6× lower in Fig. 11); thus there seems to be a large amount of improvement available if we consider architecture changes in the vehicle suspension domain. Qualitatively observing the position profiles of z_U and z_S in Figs. 6–11, we see



(a) Performance vs. percentile rank.

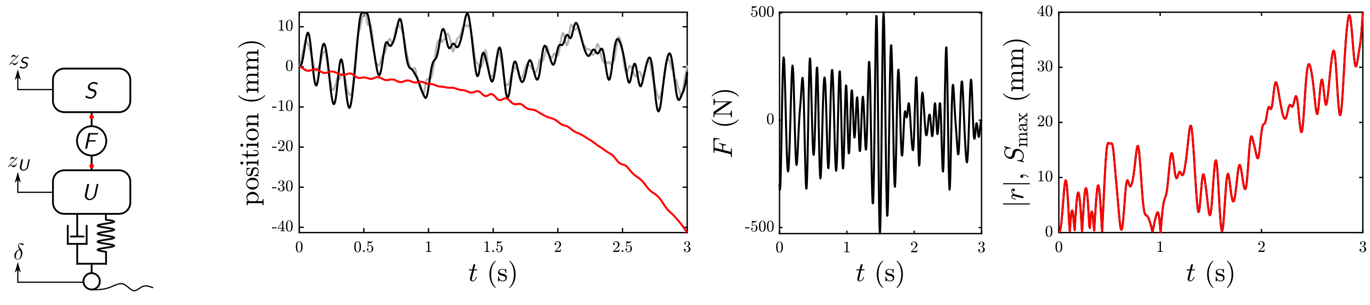
(b) Performance vs. complexity.

FIGURE 5: Summary of the results for 4,351 candidate suspensions.

TABLE 3: Objective function, plant design, and maximum force for select suspensions.

Figure	Ψ_d	handling	comfort	effort	k_1	k_2	k_3	b_1	m_1	m_2	max(F)
Fig. 6	2.32	1.00	0.42	0.89	–	–	–	–	–	–	528
Fig. 7	2.07	0.89	0.53	0.65	21.3	–	–	319	–	–	433
Fig. 8	0.91	0.51	0.13	0.27	14.1	–	–	–	3.8	–	265
Fig. 9	0.70	0.39	0.01	0.30	4.3	13.0	–	–	5.7	–	317
Fig. 10	0.52	0.30	0.07	0.15	38.3	11.7	–	–	2.8	10.0	250
Fig. 11	0.45	0.24	0.02	0.20	14.1	15.8	4.6	100	8.9	2.2	409

[*] All k have units of kN/m, b have units of Ns/m, m have units of kg, and F have units of N.



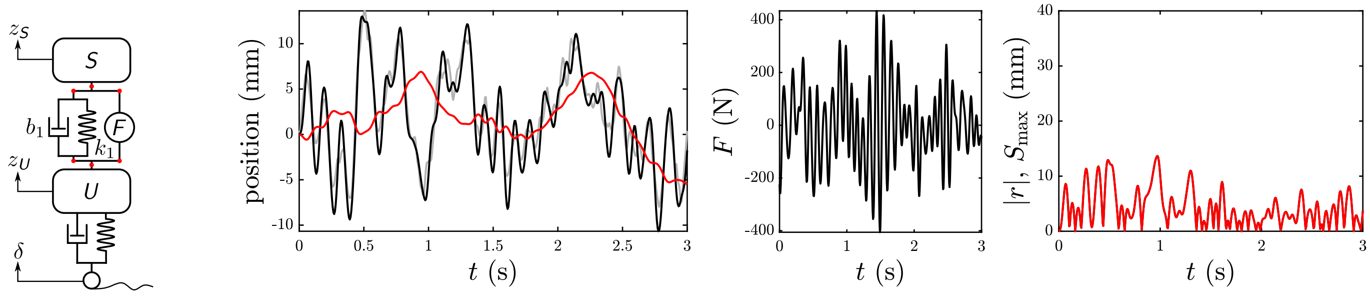
(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 6: Results for the pure active architecture ($N_c = 1$).



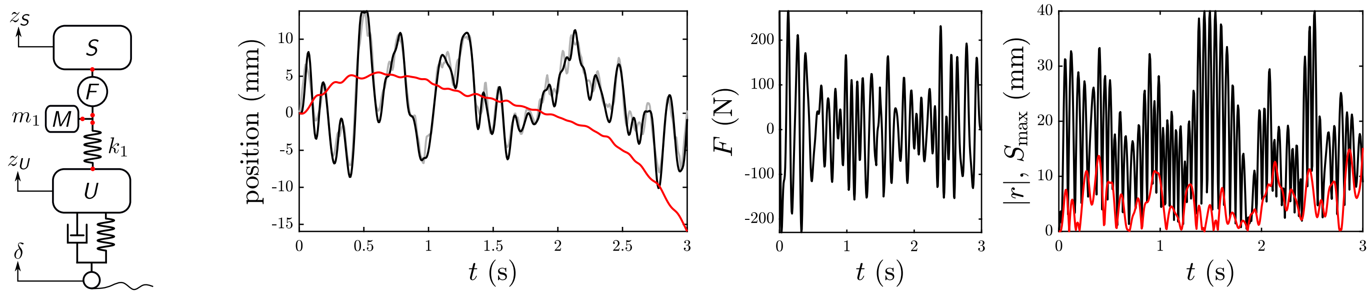
(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 7: Results for the canonical active architecture ($N_c = 3$).



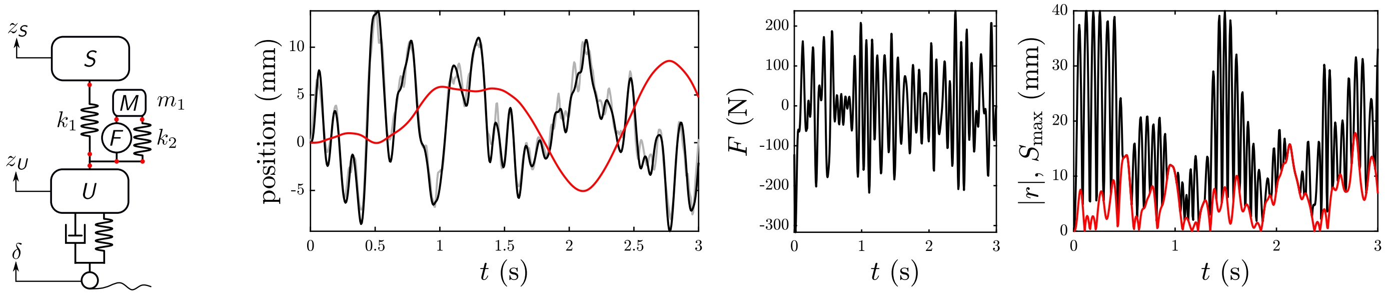
(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 8: Results for best performing architecture with $N_c = 3$.



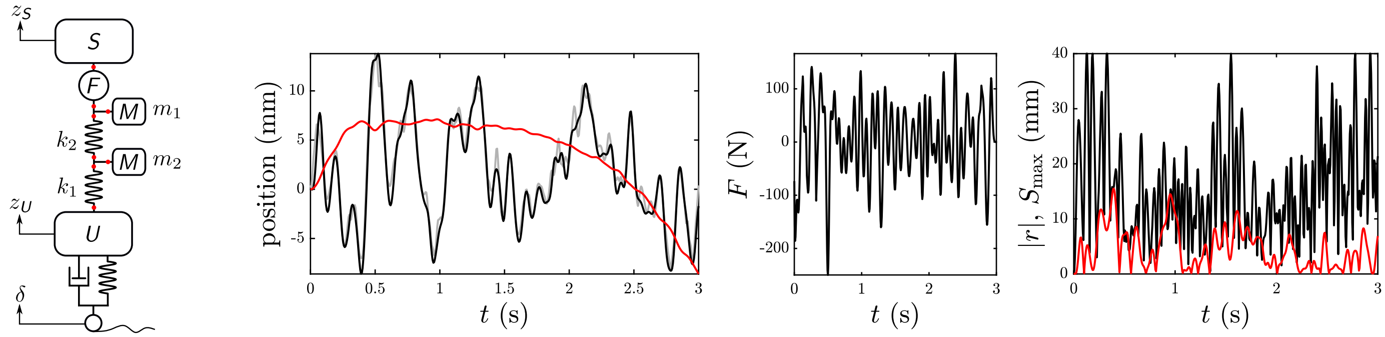
(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 9: Results for best performing architecture with $N_c = 4$.



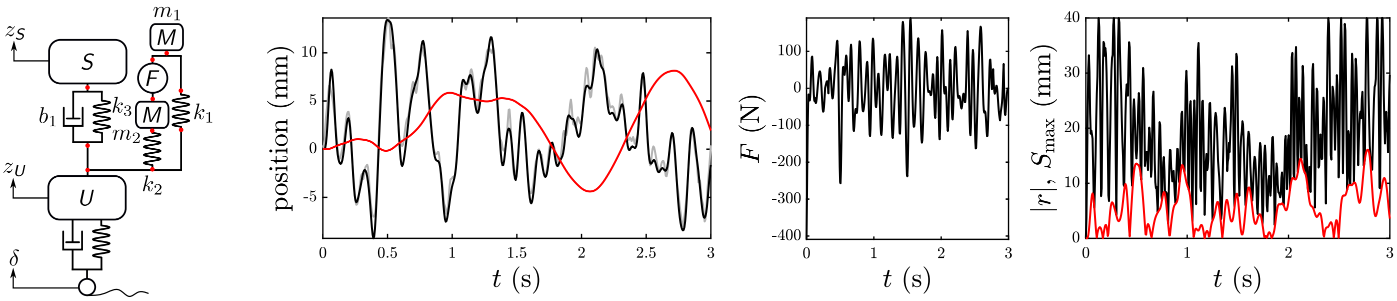
(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 10: Results for best performing architecture with $N_c = 5$.



(a) Architecture.

(b) Positions with z_S, z_U, and δ.

(c) Control.

(d) Rattlespace (r) and stroke limits.

FIGURE 11: Results for best performing architecture with $N_c = 7$.

a general smoothing of z_S as well as a better agreement between δ and z_U for the suspensions with a lower value of Ψ_d. For all the shown architectures, except the canonical active, the stroke limit is reached during some part of the time horizon, hence the value of the stroke inequality constraints in our problem formulation. The individual parts of the objective function in Eqn. (13) are also shown in Table 3. While Fig. 9 has the lowest overall objective function value, Fig. 9 has the lowest comfort objective part and Fig. 10 has the lowest control effort objective part of the shown architectures. Determining the solution to multiobjective problems will add further complexity to the design problem formulation and increased evaluation cost. It is up to further design activities to determine if the identified suspensions can satisfy other necessary design requirements that we have not considered here.

Many of the architectures have issues related to the internal positioning constraints, and thus, are not physically realizable. One of these architectures is shown in Fig. 10, where the relative positions of the springs switch during the time horizon. The primary cause of this issue is the ideal force source model for F, but other factors such as low spring stiffness can also cause overlapping elements in the suspension. Such issues with the modeling of the system were only identified during the analysis of the preliminary results. Performing this kind of architecture study can reveal representation and modeling issues that are not necessary to consider when only operating on the canonical architecture realizations.

7 CONCLUSION

In this article, a problem class with combined architecture, plant, and control design was described. Achieving successful design automation in this class of problems requires adequate theory and tools to support such considerable scope and complexity. The select problem class leverages a number of existing theory and tools and is particularly attractive due to the symbiosis between labeled graph representations of architectures, dynamic models constructed from linear physical elements, linear-quadratic dynamic optimization, and the nested co-design solution strategy. However, even with the proposed problem structure, there will continue to be challenges associated with the combinatorial nature of the architecture design, nonconvexity and feasibility of general co-design problems, and infinite-dimensional nature of dynamic optimization problems.

A combined architecture, plant, and control design problem in the described problem class was posed for a vehicle suspension application, and is an area where a limited set of architectures have been considered previously. All 4,374 candidate suspensions were evaluated through an automatically constructed co-design problem. The results demonstrated that changes to the vehicle suspension architecture can result in improved performance, but at the cost of increased mechanical complexity. In addition, there are a number of improvements that can be made to the problem formulation. The inclusion of the internal positioning constraints is essential to ensure that many of the results are physically meaningful. Multiple road inputs can also be considered simultaneously to give a better representation of all the environments in which the suspension will need to function. Finally, frequency domain properties, such as suspension quality spectral density and control energy spectral density, could also be utilized for a more effective problem formulation [10].

REFERENCES

[1] Kasturi, P., and Dupont, P., 1998. "Constrained optimal control of vibration dampers". *Journal of Sound and Vibration,* ***215***(3), Aug., pp. 499–509. doi: 10.1006/jsvi.1998.1661

[2] Hrovat, D., 1997. "Survey of advanced suspension developments and related optimal control applications". *Automatica,* ***33***(10), Oct., pp. 1781–1817. doi: 10.1016/S0005-1098(97)00101-5

[3] Hrovat, D., 1993. "Applications of optimal control to advanced automotive suspension design". *Journal of Dynamic Systems, Measurement, and Control,* ***115***(2B), June, pp. 328–342. doi: 10.1115/1.2899073

[4] Gobbi, M., and Mastinu, G., 2001. "Analytical description and optimization of the dynamic behaviour of passively suspended road vehicles". *Journal of Sound and Vibration,* ***245***(3), Aug., pp. 457–481. doi: 10.1006/jsvi.2001.3591

[5] He, Y., and McPhee, J., 2005. "Multidisciplinary design optimization of mechatronic vehicles with active suspensions". *Journal of Sound and Vibration,* ***283***(1-2), May, pp. 217–241. doi: 10.1016/j.jsv.2004.04.027

[6] Allison, J. T., 2008. "Optimal partitioning and coordination decisions in decomposition-based design optimization". Ph.D. dissertation, The University of Michigan, Ann Arbor, MI, USA, May. url: http://hdl.handle.net/2027.42/58449

[7] Deshmukh, A. P., Herber, D. R., and Allison, J. T., 2015. "Bridging the gap between open-loop and closed-loop control in co-design: A framework for complete optimal plant and control architecture design". In American Control Conference, pp. 4916–4922. doi: 10.1109/ACC.2015.7172104

[8] Koch, G., and Kloiber, T., 2014. "Driving state adaptive control of an active vehicle suspension system". *IEEE Transactions on Control Systems Technology,* ***22***(1), Jan., pp. 44–57. doi: 10.1109/tcst.2013.2240455

[9] Bourmistrova, A., Storey, I., and Subic, A., 2005. "Multiobjective optimisation of active and semi-active suspension systems with application of evolutionary algorithm". In International Conference on Modelling and Simulation, pp. 1217–1223.

[10] Fathy, H. K., Papalambros, P. Y., Ulsoy, A. G., and Hrovat, D., 2003. "Nested plant/controller optimization with application to combined passive/active automotive suspensions". In American Control Conference, Vol. 4, pp. 3375–3380. doi: 10.1109/ACC.2003.1244053

[11] Alyaqout, S. F., Papalambros, P. Y., and Ulsoy, A. G., 2007. "Combined design and robust control of a vehicle passive/active suspension". In European Control Conference, pp. 1264–1270.

[12] Allison, J. T., Guo, T., and Han, Z., 2014. "Co-design of an active suspension using simultaneous dynamic optimization". *Journal of Mechanical Design,* ***136***(8), June, p. 081003. doi: 10.1115/1.4027335

[13] Ulsoy, A. G., Hrovat, D., and Tseng, T., 1994. "Stability robustness of LQ and LQG active suspensions". *Journal of Dynamic Systems, Measurement, and Control,* ***116***(1), Mar., pp. 123–131. doi: 10.1115/1.2900666

[14] Borutzky, W., 2010. *Bond Graph Methodology*, 1st ed. Springer. doi: 10.1007/978-1-84882-882-7

[15] Kypuros, J. A., 2013. *System Dynamics and Control with Bond Graph Modeling*. CRC Press.

[16] Karnopp, D. C., Margolis, D. L., and Rosenberg, R. C., 2012. *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, 5th ed. Wiley.

[17] Gonzalez, G., and Galindo, R., 2008. "Removing the algebraic loops of a bond graph model". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* ***222***(6), Sept., pp. 543–556. doi: 10.1243/09596518jsce559

[18] Maschke, B. M., and Villarroya, M., 1995. "Properties of descriptor systems arising from bond graph models". *Mathematics and Computers in Simulation,* ***39***(5–6), Nov., pp. 491–497. doi: 10.1016/0378-4754(95)00109-1

[19] Cellier, F. E., and Nebot, À., 2005. "The Modelica bond graph library". In International Modelica Conference, pp. 57–65.

[20] Wu, Z., Campbell, M. I., and Fernández, B. R., 2008. "Bond graph based automated modeling for computer-aided design of dynamic systems". *Journal of Mechanical Design,* ***130***(4), Apr., p. 041102. doi: 10.1115/1.2885180

[21] Herber, D. R., and Allison, J. T., 2018. PM suspensions. GitHub. url: https://github.com/danielrherber/pm-suspensions

[22] The MathWorks. linearize. Accessed on 3/3/2018. url: https://www.mathworks.com/help/slcontrol/ug/linearize.html

[23] Herber, D. R., 2017. "Advances in combined architecture, plant, and control design". Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. url: https://www.ideals.illinois.edu/handle/2142/99394

[24] Fathy, H. K., Reyer, J. A., Papalambros, P. Y., and Ulsoy, A. G., 2001. "On the coupling between the plant and controller optimization problems". In American Control Conference, Vol. 3, pp. 1864–1869. doi: 10.1109/ACC.2001.946008

[25] Herber, D. R., and Allison, J. T., 2017. "Nested and simultaneous solution strategies for general combined plant and controller design problems". In ASME International Design Engineering Technical Conferences, no. DETC2017-67668, p. V02AT03A002. doi: 10.1115/DETC2017-67668

[26] Chilan, C. M., Herber, D. R., Nakka, Y. K., Chung, S.-J., Allison, J. T., Aldrich, J. B., and Alvarez-Salazar, O. S., 2017. "Co-design of strain-actuated solar arrays for spacecraft precision pointing and jitter reduction". *AIAA Journal,* ***55***(9), Sept., pp. 3180–3195. doi: 10.2514/1.J055748

[27] Deshmukh, A. P., and Allison, J. T., 2016. "Multidisciplinary dynamic optimization of horizontal axis wind turbine design". *Structural and Multidisciplinary Optimization,* ***53***(1), Jan., pp. 15–27. doi: 10.1007/s00158-015-1308-y

[28] Herber, D. R., Guo, T., and Allison, J. T., 2017. "Enumeration of architectures with perfect matchings". *Journal of Mechanical Design,* ***139***(5), Apr., p. 051403. doi: 10.1115/1.4036132

[29] Lawler, E. L., 1976. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston.

[30] Rispoli, F. J., 2007. *Applications of Discrete Mathematics*, updated ed. McGraw-Hill, ch. Applications of Subgraph Enumeration.

[31] Wyatt, D. F., Wynn, D. C., Jarrett, J. P., and Clarkson, P. J., 2011. "Supporting product architecture design using computational design synthesis with network structure constraints". *Research in Engineering Design,* ***23***(1), Apr., pp. 17–52. doi: 10.1007/s00163-011-0112-y

[32] Herber, D. R., Guo, T., and Allison, J. T., 2016. PM architectures project. GitHub. url: https://github.com/danielrherber/pm-architectures-project

[33] Herber, D. R., and Allison, J. T., 2017. Enhancements to the perfect matching-based tree algorithm for generating architectures. Technical report UIUC-ESDL-2017-02, Engineering System Design Lab, Urbana, IL, USA, Dec. url: https://ideals.illinois.edu/handle/2142/98990

[34] Herber, D. R., Lee, Y. H., and Allison, J. T., 2017. DT QP project. GitHub. url: https://github.com/danielrherber/dt-qp-project

[35] Allison, J. T., and Herber, D. R., 2014. "Multidisciplinary design optimization of dynamic engineering systems". *AIAA Journal,* ***52***(4), Apr., pp. 691–710. doi: 10.2514/1.J052182

[36] Koeln, J. P., Williams, M. A., Pangborn, H. C., and Alleyne, A. G., 2016. "Experimental validation of graph-based modeling for thermal fluid power flow systems". In ASME Dynamic Systems and Control Conference, no. DSCC2016-9782, p. V002T21A008. doi: 10.1115/DSCC2016-9782

[37] Åkesson, J., 2008. "Optimica–An extension of modelica supporting dynamic optimization". In International Modelica Conference, pp. 57–66.

[38] Patterson, M. A., and Rao, A. V., 2014. "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming". *ACM Transactions on Mathematical Software,* ***41***(1), Oct., pp. 1–37. doi: 10.1145/2558904

[39] Peddada, S. R. T., Herber, D. R., Pangborn, H. C., Alleyne, A. G., and Allison, J. T., 2018. "Optimal flow control and single split architecture exploration for fluid-based thermal management". In ASME International Design Engineering Technical Conferences, no. DETC2018-86148.

[40] Schmidt, L. C., Shetty, H., and Chase, S. C., 2000. "A graph grammar approach for structure synthesis of mechanisms". *Journal of Mechanical Design,* ***122***(4), Dec., pp. 371–376. doi: 10.1115/1.1315299

[41] Campbell, M. I., Cagan, J., and Kotovsky, K., 2000. "Agent-based synthesis of electromechanical design configurations". *Journal of Mechanical Design,* ***122***(1), Mar., pp. 61–69. doi: 10.1115/1.533546

[42] Das, A., and Vemuri, R., 2007. "An automated passive analog circuit synthesis framework using genetic algorithms". In IEEE Computer Society Annual Symposium on VLSI. doi: 10.1109/ISVLSI.2007.22

[43] Grimbleby, J. B., 2000. "Automatic analogue circuit synthesis using genetic algorithms". *IEE Proceedings - Circuits, Devices and Systems,* ***147***(6), Dec., pp. 319–323. doi: 10.1049/ip-cds:20000770

[44] Ochotta, E. S., Rutenbar, R. A., and Carley, L. R., 1996. "Synthesis of high-performance analog circuits in ASTRX/OBLX". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* ***15***(3), Mar., pp. 273–294. doi: 10.1109/43.489099

[45] Guo, T., Herber, D. R., and Allison, J. T., 2018. "Reducing evaluation cost for circuit synthesis using active learning". In ASME International Design Engineering Technical Conferences, no. DETC2018-85654.

[46] Liu, T., Azarm, S., and Chopra, N., 2017. "On decentralized optimization for a class of multisubsystem codesign problems". *Journal of Mechanical Design,* ***139***(12), Oct., p. 121404. doi: 10.1115/1.4037893